24: Inference for Regression in R

Stat 120 | Fall 2025

Prof Amanda Luby

When we are fitting a linear regression model in R, we use the `lm()` function. Here is a template:

```
lm(y ~ x, data = dataset)
```

where

- `y` is the column name for the response variable
- `x` is the column name for the categorical predictor variable
- `dataset` is the name of the data set

To get the *summary table* use

```
summary(lm_mod)
```

where `lm_mod` is the name of the model that you fit using `lm()`.

To find confidence and prediction intervals for observations, use `augment()` from the `broom` package

```
augment(lm_mod, newdata = tibble(x = ______), interval = "none", conf.level = .95)
```

where

- `lm_mod` is the name of the model that you fit using `lm()`
- `x` is the column name for the categorical predictor variable
- `_____` is filled in with the value of x that you want to predict
- `interval` is one of "none", "confidence", or "prediction"
- `conf.level` is the confidence level for the interval (default is 0.95)

We're going to use the dataset `cereals.csv`, which has nutrition data on a random sample of breakfast cereals.

1. Load in the data and check the "spreadsheet view". How many cereals are in the sample?

```
cereals <- read.csv("http://math.carleton.edu/Stat120/RLabManual/Cereals.csv")
```

2. We're going to try to predict carbs per gram (`carbsgram`) using calories per gram (`calgram`). Make a scatterplot and describe the relationship you find.

3. Find a confidence interval for the slope and report your result in context. Some starter code is below (make sure to remove `eval = FALSE`!)

```
cereal_lm = lm(________ ~ ________, data = cereals)
summary(cereal_lm)
```

4. Use the starter code below to find a 90% *prediction interval* for a new cereal with 3.5 calories per gram. Interpret your result in context.

```
augment(________, newdata = tibble(calgram = ______), interval = ______, conf.level =  ____)
```

5. The code below creates a confidence interval and prediction interval for all predicted values. One is shown in blue and one is shown in orange. Which is which, and how can you tell?

```
cereal_aug = augment(cereal_lm, interval = "prediction")
ggplot(cereal_aug, aes(x = calgram, y = carbsgram)) +
  geom_point() +
  geom_smooth(method=lm, fill = "darkblue") +
  geom_line(aes(y=.lower), color = "darkorange", linetype = "dashed") +
  geom_line(aes(y=.upper), color = "darkorange", linetype = "dashed")
```

6. There is one point that falls outside of the prediction interval. Is this evidence that it is an outlier? Should we remove it?

7. The code below makes residual plots for the `cereal_lm` model. (Remove `eval = FALSE` after you have created the model). Do you have any concerns about the LINE conditions?

```
p1 = ggplot(cereal_aug, aes(x = .resid)) +
  geom_histogram(col = "white", bins = 20)

p2 = ggplot(cereal_aug, aes(x = .fitted, y = .resid)) +
  geom_point() +
  geom_hline(yintercept = 0, col = "darkgray", linetype = "dashed")
```

```
p1
p2
```

8. When we include plots on posters or in papers, we want to be mindful of how much space they take up but also make sure they are readable. One way that I like to do this is by putting multiple plots side-by-side using the `patchwork` package. Remove `eval = FALSE` and run the code below. Next, run `p1+p2` in the *console* to make the plot show up in the *viewer*. Click on "export" and then "save as image". Change the dimensions to be 1200 pixels long x 400 pixels tall and save your plot with an informative file name. In order to get the plot from the RStudio Server onto your own computer, go to the "Files" tab and click on the name of the image. This should bring up a high-quality image that you can copy and paste into a text editor or embed directly. Alternatively, you can check the box next to the image and click on "more –> export" to download to your computer. Submit *only* this image file on gradescope (1 per group).

```
p1 + p2
```